Adaptive Smoothing Respecting Feature Directions

René A. Carmona, Member, IEEE, and Sifen Zhong

Abstract—Accurate extraction of the image feature directions is essential to image smoothing and other image processing tasks. We show that the gradient-based feature direction extraction method can be very erroneous. The gradient is too local, and it cannot detect oscillations. We have developed two new methods: the Hessian method, an approach using higher order differentiations, and the Gabor method, an approach using space-frequency analysis.

Index Terms—Adaptive smoothing, anisotropic diffusion.

I. ADAPTIVE SMOOTHING

SMOOTHING (by local weighted averaging) is an effective image regularization method that has been used for denoising, restoration, and enhancement. A drawback is that smoothing can damage image features such as edges, lines, and textures. To avoid the damage, the smoothing has to be adaptively controlled with two principles: 1) control of the amount of smoothing, i.e., less smoothing in the locations with strong image features, and more smoothing in the locations with weak image features; and 2) control of the direction of smoothing, i.e., minimal smoothing in the directions across the image features, and maximal smoothing in the directions along the image features. While both principles are crucial, this paper is mostly concerned with the principle of directional control.

A classic example of adaptive smoothing is the anisotropic diffusion scheme of Perona and Malik [7], in which the smoothing process is formulated by a partial differential equation (PDE). Given an image \( u_0(x, y) \), its smoothed versions comprise a family of images \( u_t(x, y) \), the variable \( t \) parameterizing the amount of smoothing. For \( t = 0 \), \( u(x, y; 0) \) is initialized to \( u_0(x, y) \); for \( t > 0 \), \( u_t(x, y; t) \) is obtained by solving an evolution equation:

1) Perona–Malik Equation (Div Form):

\[
\begin{align*}
\nabla u &= \text{div}[\nabla^* \mathbf{u}],
\end{align*}
\]

Here \( \nabla^* \mathbf{u} = [u_{x}, u_{y}] \) is the (spatial) gradient, and \( \nabla^* \mathbf{u} = \sqrt{u_{x}^2 + u_{y}^2} \) is the gradient magnitude. The role of the function \( \nabla^* \mathbf{u} \) is to adaptively control the smoothing. The effects of \( \nabla^* \mathbf{u} \) are better described when the smoothing is expressed in the \( \eta, \xi \) coordinate.

Let \( \eta \) denote the gradient direction, and let \( \xi \) denote the contour direction—the direction perpendicular to the gradient—also known as the direction of level set

\[
\begin{align*}
\eta &= \frac{u_{x} u_{y}}{\sqrt{u_{x}^2 + u_{y}^2}}, \quad \xi = \frac{-u_{y} u_{x}}{\sqrt{u_{x}^2 + u_{y}^2}}.
\end{align*}
\]

Let \( u_{\eta} \) and \( u_{\xi} \) denote the second-order directional derivatives in the directions of \( \eta \) and \( \xi \), respectively, as follows:

\[
\begin{align*}
u_{\eta} &= \frac{u_{x}^2 u_{xx} + 2u_{x} u_{y} u_{xy} + u_{y}^2 u_{yy}}{u_{x}^2 + u_{y}^2}, \\
u_{\xi} &= \frac{u_{x} u_{xx} - 2u_{x} u_{y} u_{xy} + u_{y}^2 u_{yy}}{u_{x}^2 + u_{y}^2}.
\end{align*}
\]

By writing out \( \text{div}[\nabla^* \mathbf{u}] \), the evolution equation can be written in terms of \( u_{\eta} \) and \( u_{\xi} \).

2) Perona–Malik Equation (\( \eta, \xi \) Form):

\[
\begin{align*}
u_t &= g(|\nabla u|) \left\{ \left[ 1 + \frac{|\nabla u|^2}{g(|\nabla u|)} \right] u_{\eta} + u_{\xi} \right\}.
\end{align*}
\]

An example of using the Perona–Malik equation is the minimize surface area scheme, in which

\[
g(|\nabla u|) = \frac{1}{\sqrt{1 + |\nabla u|^2}}.
\]

It is often implemented in the form [9]

\[
g(|\nabla u|) = \frac{1}{\sqrt{1 + \left( \frac{|\nabla u|}{\epsilon} \right)^2}}
\]

with the value of the parameter \( \epsilon \) being crucial to the performance. We shall not be concerned with this issue here because the image values can always be rescaled so that \( \epsilon \) can be omitted. The corresponding evolution equation is

\[
\begin{align*}
u_t &= \frac{1}{\sqrt{1 + |\nabla u|^2}} \left( \frac{1}{1 + |\nabla u|^2} u_{\eta} + u_{\xi} \right).
\end{align*}
\]

This typical example shows how the Perona–Malik equation handles the two principles of adaptive smoothing: 1) for the control of the amount of smoothing, the front control factor \( 1/\sqrt{1 + |\nabla u|^2} \) allows less smoothing where \( |\nabla u| \) is large and allows more smoothing where \( |\nabla u| \) is small and 2) for the control of the direction of smoothing, two uneven control factors are assigned to the \( \eta \) and \( \xi \) directions. In the \( \eta \) direction, the control factor \( 1/(1 + |\nabla u|^2) \) which corresponds to

\[
\frac{1}{1 + |\nabla u|^2}
\]

as the factor \( 1/u_{\eta} \) in (3) virtually allows no smoothing where \( |\nabla u| \) is large. In the \( \xi \) direction, the control factor is always 1, allowing maximal smoothing.
A variant of the minimize surface area equation is to drop the term of \( u_{\eta\eta} \) to disallow any smoothing in the \( \eta \) direction

\[
u_{\tau} = \frac{1}{\sqrt{1 + (\nabla u)^2}} u_{\xi\xi}.
\]

(4)

Note that this equation cannot be written in a divergence form and it is not an instance of the Perona–Malik equation. The equations of the \( \eta - \xi \) form are more general than that of the \( \text{div form} \).

The above discussion suggests that a general evolution equation for adaptive smoothing is of the form that follows.

3) General Evolution Equation:

\[
u_{\tau} = c(au_{\eta\eta} + bu_{\xi\xi}).
\]

(5)

The function \( c \) controls the amount of smoothing, and the function pair \([a, b]\) controls the unevenness of the smoothing between the \( \eta \) and \( \xi \) directions, that \([a, b] = [0, 1]\) being most uneven, and that \([a, b] = [1, 1]\) being most even. In this paper, \( a, b, \) and \( c \) are normalized with \( b = 1 \), as the usual normalization with \( \sqrt{a^2 + b^2} = 1 \) would make the Perona–Malik equation awkward to write. The functions \( a, b, \) and \( c \) should not be limited to the functions of \( |\nabla u| \), but the choice of these functions is not the main concern of this paper.

This paper is focusing on the choice of the \( \eta - \xi \) pair, which can be any estimation of across-long feature direction pair and should not be limited to the gradient-contour pair. Indeed, as shown in Section II, the gradient-contour pair can be a poor estimation of the across-long direction pair. In Sections III and IV, we give two better methods for the extraction of image feature direction: the Hessian method, an approach using higher order differentiations, and the Gabor method, an approach using space-frequency analysis. Experimental results are given in Section V.

II. GRADIENT METHOD

The anisotropic diffusion scheme uses the gradient to extract the image feature direction—the gradient direction is considered to be the direction across the image feature, and the contour direction is considered to be the direction along the image feature. This gradient method, although widely used, can be very erroneous.

Consider three models of image features—edge, line, and wave (directional texture)—defined by

\[
\text{edge}_\alpha(x, y) = \begin{cases} 
\alpha x & \text{if } |x| < \frac{1}{\alpha}, \\
\text{sign}(x) & \text{otherwise},
\end{cases}
\]

\[
\text{line}_\alpha(x, y) = \begin{cases} 
1 - |\alpha x| & \text{if } |x| < \frac{1}{\alpha}, \\
0 & \text{otherwise},
\end{cases}
\]

\[
\text{wave}_\alpha(x, y) = \sin(\alpha x).
\]

The parameter \( \alpha \) is assumed to be large—an ideal step edge has an infinitely large \( \alpha \). All the image features are of vertical direction. See Fig. 1.

The contour directions agree with the image feature directions only if there is absolutely no variation in the \( y \) direction. Suppose some small variations in the \( y \) direction are introduced by adding to these images a very smooth image, for example, a plane with a small slope

\[
\text{plane}_\beta(x, y) = \beta y.
\]

The parameter \( \beta \) is assumed to be small. Adding a smooth image does not change the image feature directions, but it does change the contour directions. See Fig. 2.

Consider the case of the edge image:

\[
\text{edge}(x, y) = \text{edge}_\alpha(x, y) + \text{plane}_\beta(x, y).
\]

The edge in the image is of direction \([0, 1]\). The edge direction is not affected by the changes of \( \alpha \) and \( \beta \) as long as \( \alpha, \beta \gg 1 \). However, the contour is of direction \([-\beta/\alpha, 1]\), which is always depending on \( \alpha \) and \( \beta \). The contour direction does not agree with the edge direction unless \( \beta = 0 \) or \( \alpha = \infty \).

The situation is much worse in the case of the wave image

\[
\text{wave}(x, y) = \text{wave}_\alpha(x, y) + \text{plane}_\beta(x, y).
\]

The pattern in the image is of direction \([0, 1]\). Changing \( \beta \) does not change the pattern direction as long as \( \beta \) is small. However, the contour is of direction \([-\beta/\alpha \cos(\alpha x), 1]\), which is very different from \([0, 1]\). At the many locations where \( \cos(\alpha x) \) is zero, the contour directions are actually perpendicular to the pattern direction.

The situation is no better in the case of the line image:

\[
\text{line}(x, y) = \text{line}_\alpha(x, y) + \text{plane}_\beta(x, y).
\]

At the peaks of the line, the contour directions are perpendicular to the line direction.
These examples show that the gradient method can be very erroneous. The adaptive smoothing schemes using the gradient method can mistakenly give maximal smoothing to the across feature direction and severely damage the image features. The image lines and textures are particularly vulnerable because the gradient method totally misses their directions. As a result, these image features are quickly destroyed. This is the major drawback of the PDE schemes that use the gradient method.

A problem with the gradient method is that the gradient is too local. The information contained in the gradient is limited to a point and its immediate neighbors, while the direction extraction should base on a larger neighborhood. Being very local can be an advantage when dealing with edges, since an edge point is a very local event. The anisotropic diffusion scheme and other PDE schemes enjoy such advantage and perform much better than the not-very-local methods such as the trigonometry and spline approximations which usually suffer from the Gibbs phenomenon. See [1], [2], [8], [12], [14], and [15] for a glance at the enormous literature about the advantages of the PDE schemes. On the other hand, the edge curve formed by a set of connected edge points is not a local event. To handle an edge curve, a less local method is needed. See [6], [10], [11], and [13] for the work on active contours. The gradient is good for the identification of edge points, but not for edge curves, which determine the edge directions.

Another problem with the gradient method is that the gradient cannot detect oscillations. Being oscillatory is the most significant property of the image features such as lines and textures. Oscillations are better detected by higher order differentiations and by space-frequency analysis.

III. HESSIAN METHOD

The Hessian method is an approach using higher order differentiations to extract the image feature direction. Analogous to the gradient method, which considers the direction of maximal (in magnitude) first-order directional derivative to be the direction across the image feature, the Hessian method considers the direction of maximal second-order directional derivative to be the direction across the image feature. Its perpendicular direction is considered to be the direction along image feature.

We use the Hessian to compute the second-order directional derivatives. Given an image $u(x, y)$, its Hessian is the matrix

$$
\begin{bmatrix}
  u_{xx} & u_{xy} \\
  u_{yx} & u_{yy}
\end{bmatrix}
$$

The two eigenvalues of the Hessian, denoted by $\lambda_1$ and $\lambda_2$, are given by

$$
\lambda_1 = \frac{1}{2} \left[ (u_{xx} + u_{yy}) + \sqrt{(u_{xx} - u_{yy})^2 + 4u_{xy}^2} \right]$$

$$
\lambda_2 = \frac{1}{2} \left[ (u_{xx} + u_{yy}) - \sqrt{(u_{xx} - u_{yy})^2 + 4u_{xy}^2} \right].
$$

Let $\lambda_\eta$ denote the largest (in magnitude) eigenvalue and let $\lambda_\xi$ denote the other eigenvalue. Let $v_\eta$ and $v_\xi$ denote the corresponding eigenvectors. The eigenvector $v_\eta$ is the direction having the maximal second-order directional derivative among all the directions. The eigenvector $v_\xi$ is perpendicular to $v_\eta$.

The direction of $v_\eta$ is considered to be the direction across the image feature, and the direction of $v_\xi$ is considered to be the direction along the image feature. Therefore, the substitutions to obtain the evolution equation are

$$
\eta = v_\eta, \quad \xi = v_\xi.
$$

Because $u_{\eta\eta}$ and $u_{\xi\xi}$ are $\lambda_\eta$ and $\lambda_\xi$, respectively, the corresponding general evolution equation takes the form

$$
u_t = c(a\lambda_\eta + b\lambda_\xi).
$$

The change from the first-order derivatives of the gradient method to the second-order derivatives of the Hessian method is fundamental. The image features are characterized by oscillations, which cannot be detected by the first-order derivatives but can be captured by the second-order derivatives. In particular, the image lines and textures generate large second-order derivatives but do not generate large first-order derivatives. The image edges generate large second-order derivatives at their sides while they generate large first-order derivatives at their centers. Overall, the second-order derivatives are much more suitable for image feature detection.

The evolution equation using the Hessian method is a second-order partial differential equation, the same order as the one using the gradient method. The computation of the eigenvalues can be done via a simple algebraic formula. Thus the computational complexity of the Hessian method is basically the same as that of the gradient method. In our implementation, all the partial derivatives are computed using central differencing schemes on $3 \times 3$ grid points. Thus the smoothing procedure is very efficient.

IV. GABOR METHOD

The Gabor method is an approach using space-frequency analysis—analogous to time-frequency analysis in one dimension [5]—to extract image feature direction. Space-frequency analysis has been exceptionally successful in dealing with image features [3], [4]. We extend its application to the extraction of image feature direction.

We use the Gabor transform—the windowed Fourier Transform—to compute the space-frequency representation of the image. Given an image $u(x, y)$, its Gabor transform $F(x, y; \omega, \theta)$ is given by

$$
F(x, y; \omega, \theta) = \int u(r, s) g(r - x, s - y) \Omega(r, s; \omega, \theta) dr ds
$$

$$
\Omega(r, s; \omega, \theta) = \exp[-i\omega(r \cos \theta + s \sin \theta)].
$$

The Gabor transform is the Fourier transform of the image after being windowed by a Gaussian window function $g(x, y)$ shifted to each location. Note that the Fourier variables $[\omega, \theta]$ are in a polar coordinate, $\omega$ being the absolute frequency and $\theta$ being the direction. At each location $[x, y]$, the Gabor transform $F(x, y; \omega, \theta)$ is the response to the frequency $\omega$ in the direction $\theta$. With the location dependency understood, we omit the spatial variable $[x, y]$ and write $F(\omega, \theta)$ as a shorthand for $F(x, y; \omega, \theta)$.

The Gabor transform possesses much richer information than the gradient and Hessian does. At each location, it
possesses information about the oscillations at all frequencies and in all directions in the neighborhood of the location. It also possesses information about the discontinuities, although the information is inconveniently spread across all frequencies. All the needed information is available, but the image feature direction is not explicitly provided by the Gabor transform. A little extra work is needed to obtain the image feature direction.

One way to utilize the Gabor transform to extract the image feature direction is to combine the information through the spectral energy, which is computed as the squared magnitude of the Gabor transform. The spectral energy is accumulated in each direction, and the direction with maximal accumulated energy, denoted by $\theta_\eta$, is obtained

$$\theta_\eta = \arg \max_\theta \left\{ \int |\mathcal{F}(\omega, \theta)|^2 p(\omega) d\omega \right\}.$$ 

The weight function $p(\omega)$ should give more weights to the high frequencies to ensure that in the $\theta_\eta$ direction the image is most oscillatory. On the other hand, the discontinuities should have higher priority than the oscillation. Thus, the high frequencies should not be overweighted. Based on the theorem that the spectral energy associated with a discontinuity decays with a order of $1/\omega^2$, the weight function should be of a order a little higher than $\omega^2$. In our experiment, we choose the weight function to be

$$p(\omega) = \omega^3.$$ 

More study is needed in choosing the weight function.

The $\theta_\eta$ direction is considered to be the direction across the image feature. Its perpendicular direction, denoted by $\theta_\xi$, is considered to be the direction along the image feature. The substitutions to obtain the evolution equation are

$$\eta = \theta_\eta,$$

$$\xi = \theta_\xi.$$ 

The Gabor method is based on more complete information than the gradient method and the Hessian method. It is less local, extracting information from a larger neighborhood. It uses information at all frequencies and in all directions, much more than the gradient method and the Hessian method that are limited to the first- and second-order derivatives. Indeed, the Gabor method can be considered as a superset of the gradient method, the Hessian method, and other potential methods using higher order differentiations. Because of the relations between the Fourier transform and the differentiations, by manipulating the weight function $p(\omega)$, the Gabor method can imitate these differentiation based methods. For example, setting $p(\omega) = \omega^2$ corresponds to the gradient method. Therefore, the Gabor method is more general and complete.

Because the Gabor method uses more information, it is more expensive in computation. In our implementation, we have to trade completeness and accuracy for efficiency. This significantly reduces the advantage of the Gabor method. An efficient implementation of the Gabor method is necessary in order to make it practically useful.

V. EXPERIMENTS

We present the results of our experiments, which are intended to illustrate the properties of the adaptive smoothing schemes using different image feature direction extraction methods. It should be emphasized that our goal in these experiments is not to demonstrate the denoising properties of the smoothing schemes. Rather, it is to show the (damaging) effects of the smoothing. In practice, the smoothing process is usually combined with an image feature preserving process, for example, the close-to-data constraint in [9]. The image feature preserving process is an important component of our current research but is out of the scope of this paper.

We run the equation

$$u_t = c(au_{\eta\eta} + bu_{\xi\xi})$$

with the same $a$, $b$, and $c$ but different $\eta$, $\xi$. In the experiments presented here, we set $[a, b] = [0, 1]$ and $c = 1/(1 + |\nabla u|^2)$, which corresponds to the popular minimize total variation scheme [9]

$$u_t = \frac{1}{1 + |\nabla u|^2} u_{\xi\xi}.$$ 

We run three versions of the equation, each using a different method to determine the direction $\xi$ and thus the term $u_{\xi\xi}$:

1) when using the gradient method, $u_{\xi\xi} = u_{\xi\xi} = (u_x^2 u_{xx} - 2u_x u_y u_{xy} + u_y^2 u_{yy})/(u_x^2 + u_y^2);

2) when using the Hessian method, $u_{\xi\xi} = \lambda \epsilon$;

3) when using the Gabor method, $u_{\xi\xi}$ is the second-order derivative in the direction $\theta_\xi$.

The image $u(x, y)$ is initialized to the input image $u_0(x, y)$. Then it is smoothed by running the evolution equation about 100 iterations until the $L_2$ distance between initial image and the smoothed image reaches a chosen value $\sigma$. Thus, when the smoothing terminates, the smoothed image $u(x, y)$ satisfies

$$\int |u(x, y) - u_0(x, y)|^2 dx dy = \sigma^2.$$ 

Normally, in an image denoising application, the value for $\sigma$ is chosen such that it is large enough to effectively remove the noise and small enough to preserve the image features. In our experiment, we intentionally choose a large value for $\sigma$ so that the smoothing effects are easy to see. The same $\sigma$ is used for all three methods, thus the amount of smoothing is the same in all three cases. The smoothing has similar effects on image noise, but quite different on image features.

Fig. 3 shows the result of an experiment on an image with the feature of waves. The hair on the forehead forms a sort of vertical wave texture. A horizontal highlight is across the hair. In the result of the gradient method, the smoothing mistakenly creates horizontal edges. It responds to the large gradients due to the highlight but ignores the oscillations that are more important to the hair. The Hessian method and the Gabor method capture the oscillations of the hair thus handle the hair and highlight properly. The oscillations of the hair are preserved better at other locations.

Fig. 4 shows the result of an experiment on an image with the feature of lines. In this image, the variations along the
Fig. 3. (a) Original image of 256 × 192 pixels. (b) Smoothing with the gradient method. (c) Smoothing with the Hessian method. (d) Smoothing with the Gabor method.

Fig. 4. (a) Original image of 256 × 192 pixels. (b) Smoothing with the gradient method. (c) Smoothing with the Hessian method. (d) Smoothing with the Gabor method.
line directions are small. Thus the errors in the computation of the line directions by the gradient method are not severe. Nevertheless, some fine lines are destroyed. The damages caused by the Hessian method and the Gabor method are significantly less.

In these experiments, the results given by the Gabor method are not as good as expected. This is due to the problem of implementation. One of the problems is that the Fourier transform is computed in a square block (7 × 7). The results are influenced by the special roles of the horizontal and vertical directions. Another problem is the coarse direction resolution—we limit the number of directions to eight.

VI. CONCLUSION AND REMARK

This paper has demonstrated the importance of the control of the direction of smoothing, one of the two principles for the adaptive smoothing scheme. We have showed that the gradient method might fail to detect the directions of the image features and we have proposed two alternative procedures for directional smoothing.

We have also mentioned the importance of the control of the amount of smoothing, the other principles for the adaptive smoothing scheme. In the anisotropic diffusion scheme, this control is limited to functions of the gradient. For the same reason that the gradient fails to detect the directions of the oscillations, the gradient fails to detect the strength of the oscillations as well. As a consequence, the anisotropic diffusion scheme cannot properly control the amount of smoothing in dealing with image lines and textures. Using higher order differentiations and space-frequency analysis to control the amount of smoothing will be the subject of a forthcoming investigation.

REFERENCES


René A. Carmona (M’92) is a Professor with the Statistics and Operations Research Program, Civil Engineering and Operations Research Department, Princeton University, Princeton, NJ. His interests include image analysis, the statistical time-frequency analysis of signals, and the theory and the numerics of stochastic processes.

Dr. Carmona is a Fellow of the Institute of Mathematical Statistics and a member of AMS, IMS, and SIAM.

Sifen Zhong received the B.A. degree in mathematics and computer science from Queens College of the City University of New York in 1985, and the M.S. and Ph.D. degrees in computer science from New York University in 1987 and 1991, respectively.

He is a Research Scientist in the Statistics and Operations Research Program, Civil Engineering and Operations Research Department, Princeton University, Princeton, NJ. Since his fundamental work on the characterization of signals and images by the extrema of their wavelet transforms, he has contributed several papers to the wavelet transform applications and the PDE approach to image coding and image processing.